Issue 30 | Oct 2010

Blender learning made easy

blender art MAGAZINE

Once Upon an Image

The Parent Inverse and the Origin of Children

Does Your Character Have Any Feelings

7 Elements of Digital Storytelling

The Importance Of Bodylanguage

COVERART Confinement - by reynante

blender art MAGAZINE

EDITOR
Gaurav Nawani gaurav@blenderart.org

MANAGING EDITOR
Sandra Gilbert sandra@blenderart.org

WEBSITE
Nam Pham nam@blenderart.org

DESIGNER
Gaurav, Sandra, Alex

PROOFERS
Brian Treacy
Daniel Hand
Daniel Mate
Fade Shayol
Joel Godin
Joshua Leung
Joshua Scotton
Marion Piper
Mark Warren
Patrick O'Donnell
Phillip Ryals
Scott Hill
Tansunn Kitsuki
Valérie Hambert
Wade Bick

WRITERS
dreamsgate
Pep Ribal
Aaron Powell
Marc Leboeuf

COVER ART
Confinement - by reynante

CONTENTS

Sandra Gilbert
Managing Editor

"Stories are easier to tell in animation form than in just a single image. But even though they are easier to tell, they require an enormous amount of work, planning and dedication on the part of the storyteller. "

I have been an avid reader ever since I was old enough to pick up a book. Getting lost in a story is one of my greatest pleasures. And even worse, I love to tell stories, just ask my family. I can, and do, go on and on and on. Whether in written form or any number of available digital formats (i.e. images, animations, presentations etc.) storytelling is an important part of the human psyche. From the dawn of man, stories have been told, shared and embellished. Stories have been, and still are, a valuable way to teach important life lessons and share experiences that evoke strong emotions and a sense of connecting or belonging. They can also be a great form of entertainment.

The ways stories have been shared has changed since the first story was told, but still they are told and retold with ever new interpretations and variations. In the world of CG art, stories are most easily told in animation. Let's face it, subjects move, plots move forward, conclusions/climaxes are reached and the story ends. Stories are easier to tell in animation form than in just a single image. But even though they are easier to tell, they require an enormous amount of work, planning and dedication on the part of the storyteller.

Capturing a story in a single image can be difficult to say the least. Do you show the action about to take place, already done or somewhere in between. And how do you show action about to take place? A single image presents challenges not found in animation, yet many of the same principles still apply when setting it up. Good composition/staging, lighting, emotions, action (okay suggestion of action) are needed elements for both single images and animations. Getting it right can make the difference between just another pretty image and a meaningful image that sticks with the viewer.

We are going to take a look at the various elements that make for memorable images and animations. So gather round blender kiddies as we begin our tale...

Once upon... an image?

IZZY SPEAKS : Sintel

"I was watching Sintel with my 4 year old again for the dozenth time in 4 days"

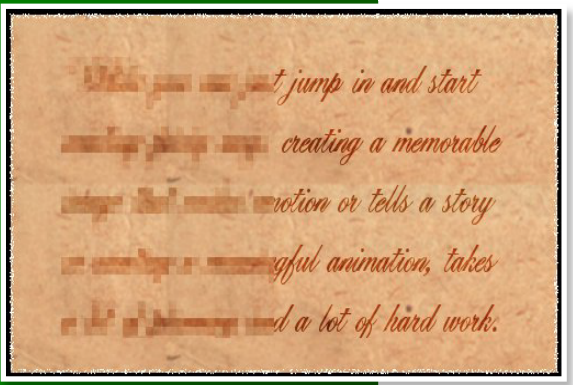
Unlike many other viewers, I have seen Sintel on a daily basis since its online release. Due in large part to extreme excitement blocking out a totally foreseeable realization. I should have known that letting my 4 year old see it with me was going to trigger massive marathon viewings of Sintel. Which of course it has.

All it took was her seeing it once, for her to immediately fall in love with the characters and the story. I must admit that mommy was rather impressed with the first viewing as well. It is a beautifully directed story that draws the viewer in. The music was amazing and tied everything together so nicely. I was so proud of the Durian team and how well they did.

And yet, I have to admit that over the years I have become jaded when watching animations. While part of me gets lost in the story, there is this analytical little voice yattering away in the background, pointing out all the little boo boos and technical glitches that plague even the most well polished commercial offerings. So yes, I did see the places in Sintel where things could have been polished and tweaked further. They were bound to be there, it happens.

And yet something rather amazing happened about 4 days after the release. I was watching Sintel with my 4 year old again for the dozenth time in 4 days. As I was sitting in front on my computer with her curled up in my lap, she looked up at me with the most amazing expression. While reaching out and grabbing my hand, she turned back to Sintel to become mesmerized once again as the story unfolded. At that moment I realized that somewhere over the last 4 days, that annoying little analytical voice had fallen silent. I was able to simply sit and enjoy a beautiful story with my little girl.

As I sat there, I realized that the Durian team had done far better than anyone had given them credit for. The whole point of stories is to create bonds that can be shared. Most often this is accomplished between the viewer and the characters. But sometimes if you are really good, you manage to create bonds between the people watching it



by dreamsgate

Blender has become a powerful tool for creating and sharing artistic visions. Over the last several years a number of excellent images and animations have been created by very talented members of the community. The increasing quality of artwork produced and of Blender itself has served as an inspiration for many to jump into 3D for the first time.

While you can just jump in and start creating fairly easily, creating a memorable image that evokes emotion or tells a story or creating a meaningful animation takes a bit of planning and a lot of hard work. During the planning stages of your project, you need to consider what you are trying to express and share with your project. Are you just creating a one off image to showcase skills learned or are you actually trying to communicate something important?

The Center for Digital Storytelling (CDS) in Berkeley, California is known for developing and disseminating the Seven Elements of Digital Storytelling, which are often cited as a useful starting point as you begin working with digital stories. CDS' Seven Elements of Digital Storytelling are geared for digital animations and presentations, but can be useful when creating a single image that sends a powerful message as well.

Okay, so just what are the Seven Elements of Digital Storytelling?

1. Point of View: What is the main point of the story and what is the perspective of the author?
2. A Dramatic Question : A key question that keeps the viewer's attention and will be answered by the end of the story.
3. Emotional Content : Serious issues that come alive in a personal and powerful way and connects the story to the audience.
4. The Gift of Your Voice : A way to personalize the story to help the audience understand the context.
5. The Power of the Soundtrack : Music or other sounds that support and embellish the storyline.
6. Economy : Using just enough content to tell the story without overloading the viewer.
7. Pacing : The rhythm of the story and how slowly or quickly it progresses.

*__note:__ when creating a single image that tells a story, numbers 5 & 7 don't really apply.

http://digitalstorytelling.coe.uh.edu/7elements.html

CDS's Digital Storytelling Cookbook - February 2007 (See Chapter 2)

http://www.storycenter.org/cookbook.pdf.

This diagram will help you picture the role that each of the 7 Elements of Digital Storytelling play in creating memorable stories ●

by dreamsgate

ARTICLE: Does your Character Have any Feelings?

by dreamsgate

We have all seen them- beautifully modeled and textured characters that have no depth. They look like little actions figures. Which don't get me wrong, action figures are cool. But they are posed with one expression and generally one body pose. Which might be totally awesome for that one instance it was designed for, but does little for further exploration of a character or storyline. In order to connect with a character, the viewer needs to see that the character actually has a full range of emotions.

Of course the best way to show this is through well thought out facial expressions that can be easily read. But even great facial expressions by themselves aren't going to do it. The facial expression needs to reflect the body language and what is actually going on in the scene. Nothing is more confusing for a viewer than a character bouncing around like a happy kid in a candy store with a sad, woe begone facial expression, in a scene that doesn't call for either.

Now, obviously that was a very silly example, but it does illustrate the point that facial expressions and body language actually need to reflect what is happening to the character in the scene.

Just how do we go about creating believable facial expressions? Well, having a limited budget, I usually turn to the Internet when I want to know something. It is absolutely amazing what you can find.

While buzzing around online (something I do far too much of), I ran across a very cool blog run by Dani Jones. Oddly enough, the link I followed dropped me right on the best post ever, 50 Facial Expressions and How to Draw Them. Now granted, he is a traditional illustrator and we work in 3d. But the knowledge he shared in this post transfers beautifully from 2d to 3d. I encourage you to go check out his blog (http://danidraws.com/blog/).

In the meantime, here are some important tips he posted about facial expressions:

The Most Important Features

1.The Eyes – Probably the most important feature for evoking a clear emotion. Utilize the eyelids and eyebrows to create your effect.

2.The Cheeks – The way they squash and stretch will affect the look and position of the eyes.

3.The Mouth – The shape of the mouth is also very important. It affects how the cheeks move and the shape of the entire face.

Additional Tips

Note that when you move the shape and position of one feature, it affects everything else. Nothing stands completely on its own.

For a stronger drawing and character, really push the expression. Instead of simply drawing a happy person, draw one that is ecstatic; instead of drawing an angry person, draw a furious one.

Have a mirror nearby. When I'm trying to nail down an expression, I often find my own face making weird movements unconsciously. It can make for good reference.

In addition to the above tips, Dani Jones also had a Facial Expressions Chart of 50 different facial expressions for you to study and a pdf for download. Some of the expressions are just priceless and are well worth checking out.

Now you have some nice facial expression reference, what next? Well there is no one better to consult than the masters. Further exploration led me to http://www.frankanollie.com, online home of Frank Thomas & Ollie Johnston. They have a great page on their site (actually their whole site is filled with great information) that discusses emotion in animation. They developed a list of 12 questions that you should ask yourself when animating your characters.

Keys to Emotion in Animation

12 QUESTIONS

by Frank Thomas & Ollie Johnston, June 2003

1 Is the character doing what the director wants in the sequence?

2 Is the character doing only one thing at a time?

3 Is the character putting over the story point in the scene you are doing?

4 Is the character acting as if there is something going on in his mind?

5 Does the character appear to be doing something on his own?

6 Can the audience tell what the character is thinking?

7 How does what the character is doing effect what the audience is thinking?

8 Does the character have appeal?

9 Is it passionate? Is passion going into the drawing and coming out of the character?

10 Is it the simplest way to do it?

11 Have you made small story sketches of one important character to be sure everything is working before you make a lot of drawings?

12 Would any one else besides your mother like what you have done?

Now we have a bit of reference and direction, and it is time for the Blender component. How do we get those wonderful expressions? Of course there is not just one way to do anything in Blender so here is a list of video tutorials that cover various techniques you can use to create and animate facial features.

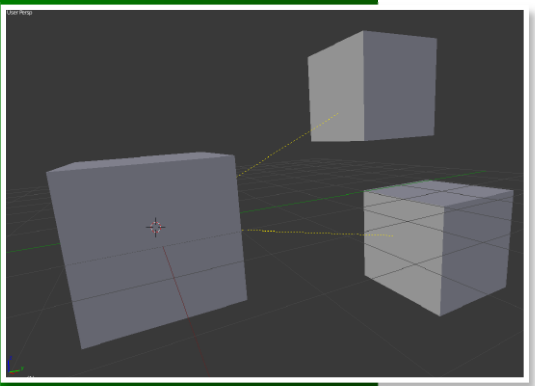
Rigging a Pupil for Dilation

Creating a Face Rig

Learning Action Constraints

Time to practice, practice, practice. Remember to look in a mirror often and watch your own face, get your buddies to make faces for you as well. In no time you will be creating an expressive range of emotions

by dreamsgate



By Pep Ribal

Introduction

This is not an article about where babies come from. Actually what I'm talking about is child objects, their location, rotation and scale in relation to their parent objects.

I want to write about this topic that drove me nuts when I started learning Blender. I'm talking about the obscure and terrifying Parent Inverse. The first thing that came to my mind when I read about it was my father hanging upside down. But I consider myself a good son, so I quickly wiped that image off my mind.

I found it really confusing at the beginning: it was very difficult to understand what exactly was the Parent Inverse, and why the location coordinates of the child object remained the same after aligning it to the position of its parent with Alt-O (Clear Origin).

At that time I had used other 3D applications, in which the origin of a child object was always the location of its parent, as plain as that. But in Blender things seemed to work a bit different, and I couldn't find out exactly how, even in all the documentation around. It is still difficult to find that explanation out there. Even when I posted the question in Blender forums a time long ago, all I got were replies of the type "I don't exactly know, but..." or "I'm also wondering how it works, but...". I was even asked "why on earth I wanted to know about all that stuff". So that's why I'd like to shed some light onto this obscure issue.

First, we will focus on location coordinates for this first example, so scale and rotation will be left apart for now.

Let's start with the following scenario: just one cube and a UV sphere. Let's say the cube is located at the global point (3,3,0), and the sphere at global point (5,5,0). A Top Ortho view will be very useful for this experiment.

What happens if we make the cube the parent of the sphere? RMB the sphere, then Shift-RMB the cube, press Ctrl-P, and select Object. Now the cube is the parent of the sphere. Neither of the objects have moved apparently. So, what are the coordinates of both objects now? Well, actually the same as before (you can check in the object transform properties, hotkey N). But the real question is: what is the origin of both objects now?

This is an easy question. Provided that no object moved, and there wasn't any change in location coordinates, the origin should be the same as before: the global origin (0,0,0).

Does it make sense?

Well, for the parent cube, it definitely does, as it's still a global object (it has no parent). But, shouldn't the origin of the sphere be the location of its parent? Actually it is, but with a small modification. If we now move the parent cube to (4,4,0), what happens to the sphere? It is apparently at global (6,6,0)... And its location (local) coordinates haven't changed; they still are (5,5,0), as the Properties sidebar shows. That means that its origin is now (1,1,0). Why?

by Pep Ribal

It's obvious that the sphere origin moves with the parent cube, but it's not exactly the location of the parent cube. When the cube was at (3,3,0), the child origin was at (0,0,0); now the cube is at (4,4,0) and the child origin is at (1,1,0)...

So, we can easily see that the child origin is the parent minus (3,3,0), or the parent origin plus (-3,-3,0), which will give the same result. What is this (-3,-3,0) value? Well, that is exactly the Parent Inverse...!

To speak properly, that is not exactly the Parent Inverse. So before we proceed, we need now to introduce briefly the concept of Transformation Matrices.

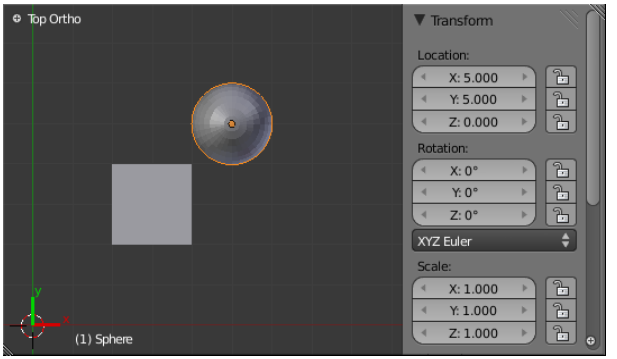
The Matrix

Even if you are not the One (and you are probably not, don't fool yourself), you deserve to know that location, rotation and scale of any object, in each of the 3 axes (X, Y and Z) are stored internally in a matrix of 4x4 numbers, called the Transformation Matrix of the object. The contents of the matrix of the active object are shown in the Transform panel of the Properties sidebar in the 3D View (hotkey N) while in Object Mode, translated to easily understandable coordinate numbers that show the transformation values of the active object in the 3 axes.

Every object has its own associated transformation matrix. To know the effective location, rotation and scale of a given object, we need two things: its transformation matrix, and its origin (the departing point of that transformation). For a global object (that has no parent), this origin is location (0,0,0), rotation (0,0,0), and scale (1,1,1). For a child object, this origin is its parent location, rotation and scale, but child objects also have an additional matrix applied: the Parent Inverse.

So we need to introduce the concept of "inverse matrix". What is an inverse transformation matrix? It's a matrix that when applied to an object takes it back to its origin. For instance, given a global object at its origin, if we apply a series of transformations on it (location, rotations and scaling), that global object ends up having a transformation matrix that shows all these transformations. The inverse of this matrix, applied to the same object undoes all of them, and the object rests again in its origin, with no rotations or scaling at all.

To make it simple, let's go back to the cube/sphere example. Let's focus on location only. At the very beginning, when the sphere wasn't related to the cube yet, the cube was at global (3,3,0). Let's simplify things, saying (though it's wrong) that the transformation matrix of the cube was "location (3,3,0)". The inverse transformation matrix, the one that would take the cube back to its origin, is naturally "location (-3,-3,0)", as (3,3,0) + (-3,-3,0) = (0,0,0), that is, matrix + inverse matrix = origin.



So, now we know what the inverse transformation matrix of the cube is. Blender is not using this inverse

matrix yet, but as soon as the sphere becomes the child of the cube, that inverse matrix (of the parent cube) is calculated and effectively applied to the child sphere. That's how we find the real origin of the sphere.

So to summarize: *at the moment of parenting, the matrix that would take the parent to its origin (the parent's inverse transformation matrix) is calculated and applied to the new child.*

As you might remember, the effective origin of the sphere was the location of the parent object, plus "location (-3,-3,0)". That is, the location of the parent plus the Parent Inverse. So why does Blender act this way? Why does it use the inverse transformation matrix of a parent object into child objects? Let's actually see what would happen if Blender didn't do so. There are 2 options:

a What if Blender parents the sphere to the cube without applying the Parent Inverse to the sphere, and without changing the transformation matrix of the sphere child? That is, without changing its location, rotation and scale values. This means that as the origin of the sphere is changing from the global origin to the parent object location, the actual sphere would change its position visibly on screen. Thus, with the position of the sphere being (5,5,0), and its new origin (3,3,0), we would automatically see the sphere jump to (3,3,0) + (5,5,0), that is, (8,8,0).

b What if Blender wants to avoid this jump at the moment of parenting, and still not apply the Parent Inverse? Then the actual location coordinates of the sphere would need to be changed. In this case, as the new origin is (3,3,0), the new location coordinates of the sphere should be changed to (2,2,0) so that it remains at global (5,5,0), as (3,3,0) + (2,2,0) = (5,5,0). But that wouldn't be too suitable, as we would be changing the object attributes (transformation matrix) for the sake of parenting, which is not justified. What would happen to such a sphere if we cleared the parenting relation with the cube, or we deleted the cube? It would jump to (2,2,0), which is bad. So this one is not an option.

That said, the only way to preserve the object attributes (its own transformation matrix) and yet avoid the jump when parenting, is to apply the Parent Inverse to the child.

So to summarize all this, the **global location of the child** is: *the location of the parent + the Parent Inverse as it was at the moment of parenting + the (local) location coordinates of the child*.

We speak of global coordinates when they are relative to the global origin; local coordinates are relative to some parent object. So the transformation properties of an object become local as soon as we parent the object, and they are global as soon as we unparent it.

So, to make it easier to understand we have focused exclusively on location, but as I mentioned earlier, the parent inverse transformation matrix (the Parent Inverse) is made up of all three types of transformations: location, rotation and scale. There is no need to go over the example again focusing on these other transformation types, as they work in a similar way. If the parent has a rotation of (10,-40,90), the Parent Inverse matrix will have a rotation of (-10,40,-90); if the parent's scale is (2,1,2), the scale of the Parent Inverse applied to the child will be (0.5,1,0.5), and so on...

Keep in mind that as soon as the parenting is done, the Parent Inverse matrix applied to the child is the inverse of the parent's transformation matrix at the very moment of parenting; that is, that matrix is never changed

by Pep Ribal

afterwards, even if we apply one thousand transformations to the parent. There is a way to change that matrix applied to the child as you will see next.

Another question to consider is that a global object, an object that has no parent, doesn't have a Parent Inverse applied, naturally. This only affects child objects.

Clear Origin (Alt-O)

In the sphere/cube example, what will happen if we clear the sphere location (**Alt-G**)? Its location coordinates will go (0,0,0), and so it will jump to its origin. If you remember, that origin is exactly its parent cube location plus the cube inverse matrix at the moment of parenting. This means that the sphere will jump to the position of the parent plus (-3,-3,0).

In Blender, with the child object active, you can press **Alt-P** and select Clear Parent Inverse. What does that do? Well, it clears the Parent Inverse... Surprised? Me too.

In this case (with location already cleared) that means that the sphere jumps to the same location of its parent. When the Parent Inverse is cleared, the origin of the child is actually the parent location, rotation and scaling, as simple as that. The Parent Inverse is ignored from then on, unless we modify it.

Clear Parent Inverse doesn't have any effect on non-child objects.

However, there is another way to make the child jump directly to the location of the parent without clearing the child location and the Parent Inverse. It's the command Clear Origin (**Alt-O**). This command only affects location (not rotation or scale). It makes the child object jump to the same global position of its parent, so that we see them placed in the same global coordinate.

And it does it without changing the child's attributes: it actually changes the values of the child's Parent Inverse matrix accordingly which will no longer have the value calculated at the time of parenting. This is the only way to change the Parent Inverse matrix of a child object in Blender (unless you use a Python script to change its values, naturally).

Unparenting (Alt-P)

We have already seen the **Clear Parent Inverse** (**Alt-P**) command. There are other uses for Alt-P. None of these have effect on global objects:

The first one is **Clear Parent**, which will cancel the parenting relationship between the selected object(s) and his (their) parents. The effect of the transformations of the parent will be discarded, and so the formerly child object will jump according to its new (global) origin, in relation to location, rotation and scale. Its local coordinates will become global but unchanged in value.

The other one is **Clear and Keep Transformation**. This one also cancels the parenting relationship, but it changes the child object attributes (location, rotation, scale) so that the transformations of the parent are applied to the child and when the child becomes a global object, no apparent change is seen on screen. In other words, it translates its local coordinates to the corresponding global ones.

Parenting methods

Besides the usual **Ctrl-P** parenting command, there is another one: Make Parent without inverse (**Shift-Ctrl-P**). This command is equivalent to making a usual parenting (**Ctrl-P**), then clearing the Parent Inverse (**Alt-P**), and finally clearing the child's location

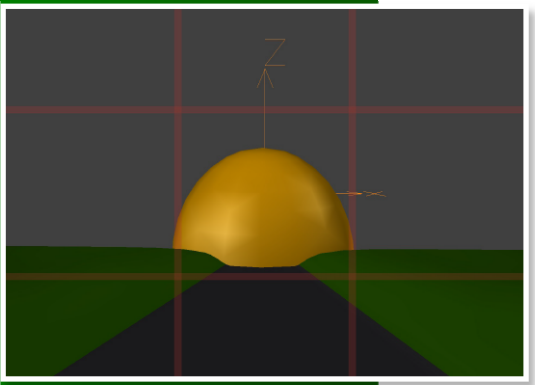
by Pep Ribal

(just the location, with **Alt-G**). The child jumps into the same global point where its parent is, without Parent Inverse matrix applied, and with its transform properties intact, except for its location coordinates which become (0,0,0).

And that's pretty much all, folks. I hope to have made this topic a bit more clear, as it was a difficult thing to grasp, in my case. But it's known that things between parents and children are always very difficult ●

Be good!

Pep Ribal works as an IT Engineer. However he is very interested in the audiovisual and multimedia world, and he has worked in a few TV productions and short films as a director, actor, screenplay writer and video editor. He has made a few small 3D works for TV using Blender.



by dreamsgate

Introduction

I had this wonderful article all planned out on the Rule of Thirds, something we should all know about and have been practicing. To explain my project, I needed a Rule of Thirds grid in Blender. Now I remembered reading in Roger Wickes' book "Foundation Blender Compositing", that he routinely used a Rule of Thirds grid to line up his projects. So I grab his book, only to realize that I seem to have misplaced the DVD that goes with it.

No problem, it can't be that hard to set up. So I fire up blender and make a grid, line it up with my camera, set it to wire and... nothing. No grid lines. Obviously I have done something wrong. Where is that disc? More looking, still can't find it. Hmmmm. Okay, I go to the Friends of Ed website and download all the tutorial files. Boy that was a big download. A quick search through the files gets me to the right blend file, but this really isn't my day, because I can't for the life of me find the promised grid. Grrrrrrrrr.

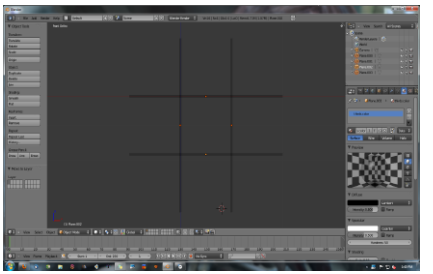
Okay, Roger and I can't be the first to want a grid in blender, so I run a google search. Yay, Andrew Price covered that on his site BlenderGuru. Awww, nuts his technique would be helpful in many situations, but not for what I had in mind. This shouldn't be this hard. Okay, I am a fairly smart woman, and Blender always provides more than one way to accomplish anything. I just need to sit down and think about this.

What I want is a visible grid parented to the camera that I can use to line up my shot. So after a little brainstorming I came up with a simple idea that should work just fine.

Rule of Thirds Grid:

- Add a plane and re-size it so that it is long and rather skinny.
- Duplicate it and move it over (hold down control and move it one major grid unit)
- Duplicate both planes and rotate them 90 degrees and line them up on the grid (fig 1)
- Join all the planes into one object
- Go into camera view and re-size the grid to fit in the camera view
- In side view, move the grid object closer to the camera, re-size if necessary and parent to the camera.

Now when ever and where ever you move the camera, the grid object will automatically move as well.



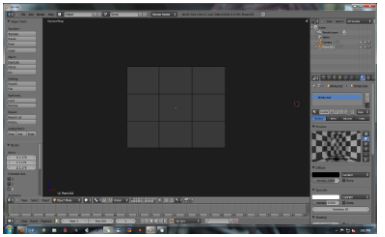
Additional tips:

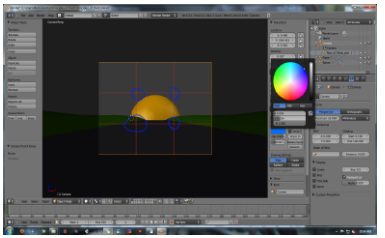
In the outliner window, toggle off the "restrict renderability" icon (the little camera). Now the grid object will not render in your final image.

by dreamsgate

You can add a simple material, I chose a basic black or depending on the scene red might be better, set to 0.250 alpha, then checked the Transparency option in the Display settings of the Object properties panel. This way you can see it, but it won't overly interfere with you scene visibility.

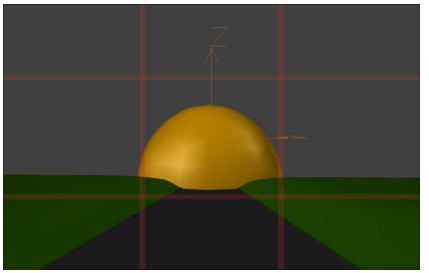
Alright, now I have a grid and I can finally get on with my article. So we all know that there are four main "power points" when using the Principle of Thirds. (fig 3) A viewer's eye is automatically drawn to whatever lies in that magical area. By controlling what is framed there, we can focus our story image and the message received by the viewer.



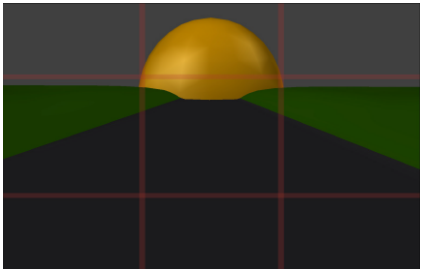


I ran across a very cool website- visualstorytelling.com, for the book, Visual Storytelling by Anthony C. Caputo, (which is unfortunately out of print.) On his site he had a great example that explained the Principle of Thirds. While it is a very simple example, it does show clearly how simply moving what is contained in the power center changes the whole message of an image.

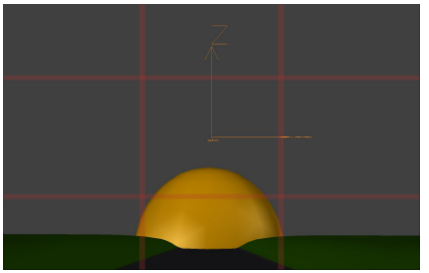
I modeled a quick simple scene to illustrate how easy it is to change the focus of your image and what will catch your viewer's attention first.



This first one is all about the sunset (sunset image).



The second one is all about the road (road image).



And the last image is all about the sky (sky image).

All three images have exactly the same elements, yet tell different stories due to how each one is framed.

In Blender it was very easy for me to set up a Principle of Thirds framing rig to create these examples. Once I had created my Principle of Thirds camera grid, I simply added an empty and made the camera track the empty. Now it was too easy to adjust my framing to achieve just the effect I was going for.

Hmmmm, think I'm going to save that rig for future use

by dreamsgate

ARTICLE: Zoybaroque

From the Blenderart Staff! We are so excited to announce that Marc Leboef from Montreal, won the Fantasy category of the The Zoybar–Shapeways-Blender challenge.

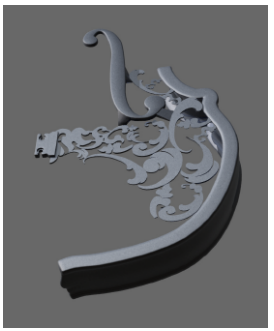
Marc: I am a freelancer and teacher in 3D. I also teach special needs kids (mostly autistics) with Blender.

When I first saw a Zoybar guitar kit, I first noticed the post-modern look of the instrument. Traditional instruments are made of wood, which give them warmth and soul. That is how the Zoybaroque design came to my mind. I wanted to give warmth and soul to the hardware.

Thinking metal instead of wood, I start searching for references of the baroque era. I made the model by thinking like an iron crafter. I wanted to show F holes and a headstock, like traditional stringed instruments. I designed the shapes with a vector application, imported the lines and started the modeling process.

For the headstock, I started the volute using the Screw Tool. I used Sculptris instead of Blender's own sculpting tools for the lion.

To keep things in proportions and lined up, A photo of myself was taken playing a... broom!



by Marc Leboeuf

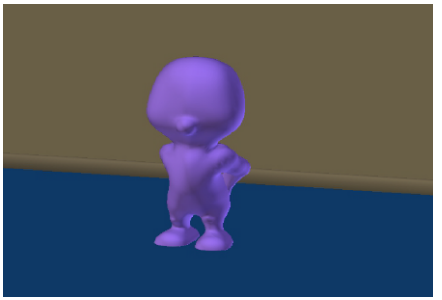
by dreamsgate

Never underestimate the importance of body language when posing a CG/3D character. Poor posing of your character can muddy the message you are trying to send to the viewer. In an animation you might get away with it to a certain extent, depending on other factors and elements in the project. But a still image with confusing or conflicting body language will be unable to convey the proper message.

I found a great list of behaviors and meanings online that can help by giving you a starting point, when deciding how to pose a character. Now some of these are definitely geared towards animation and movement. But many can be posed for a still image or in the middle of a movement, say mid step in the “Confidence = brisk, erect walk”

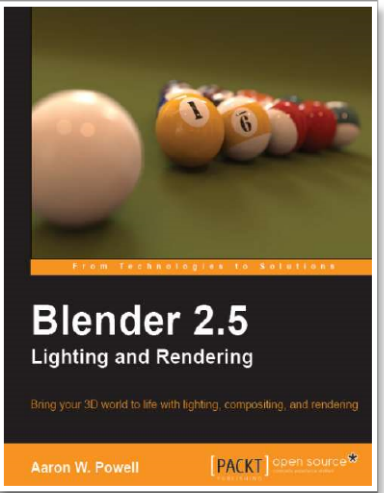
Examples Of Body Language

And since images are more fun than just descriptive text, here are a few of the more fun ones that I played with. Granted, having a character that actually has a face, will help a lot in conveying the right message. Emotions go hand in hand with body language and give important visual clues as to what is going on.



NONVERBAL BEHAVIOR	INTERPRETATION
Brisk, erect walk	Confidence
Standing with hands on hips	Readiness, aggression
Sitting with legs crossed, foot kicking slightly	Boredom
Sitting, legs apart	Open, relaxed
Arms crossed on chest	Defensiveness
Walking with hands in pockets, shoulders hunched	Dejection
Hand to cheek	Evaluation, thinking
Touching, slightly rubbing nose	Rejection, doubt, lying
Rubbing the eye	Doubt, disbelief
Hands clasped behind back	Anger, frustration, apprehension
Locked ankles	Apprehension
Head resting in hand, eyes downcast	Boredom
Rubbing hands	Anticipation
Sitting with hands clasped behind head, legs crossed	Confidence, superiority
Open palm	Sincerity, openness, innocence
Pinching bridge of nose, eyes closed	Negative evaluation
Tapping or drumming fingers	Impatience
Steepling fingers	Authoritative
Patting/fondling hair	Lack of self-confidence, insecurity
Tilted head	Interest
Stroking chin	Trying to make a decision
Looking down, face turned away	Disbelief
Biting nails	Insecurity, nervousness
Pulling or tugging at ear	Indecision

by dreamsgate



by Aaron Powell

Over the past year and a half I've had the tremendous opportunity to work with Packet Publishing on "Blender 2.5 Lighting and Rendering", a book that aims to not only help beginners get into the field of lighting and rendering in 3D, but to help freelancers and professionals enhance the quality of their renders by learning how to approach their scenes in new ways. Over the course of the book, readers learn how to light three common types of environments: exterior scenes, interior scenes, and "hybrid" scenes that contain both natural and artificial light sources.

The most rewarding aspect of the writing process was how much I learned over the course of the book's development. As an artist I truly believe that the only way to truly understand a concept is to teach it and writing "Blender 2.5 Lighting and Rendering" is a testament to that. The editors at Packt Publishing were amazingly patient as we worked - and reworked - the content over the course of the first few months.

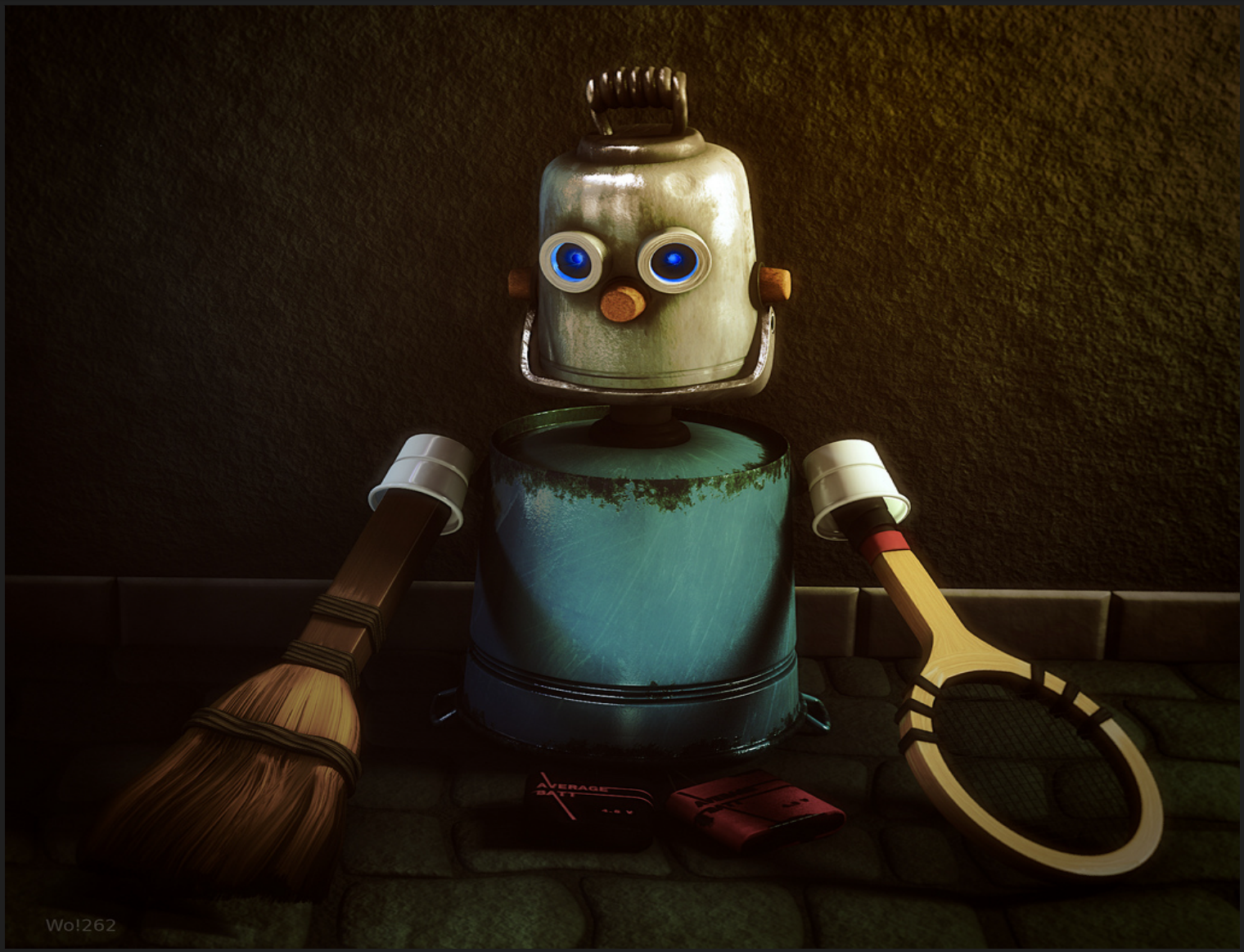
After developing an initial outline for the book, we started writing first drafts, but soon realized that the approach we were taking wasn't ideal, and it didn't engage the reader as much as we had originally hoped. Five months into the project, we decided the book had to be restarted - from scratch. Running full-tilt back to the drawing board, I developed a new outline that soon became the road map for the final book. Instead of leading the reader through Blender's wide array of rendering features, the new outline proposed that the book would explain how to approach three environments commonly found in production, what to look for, and how to use Blender's lighting and rendering features to achieve the desired results. These "environment types" included exterior environments, interior environments, and environments that had both exterior as well as "interior" light sources.

With the new outline and new goal, the writing process resumed. After entering the final draft stages, production sped up from an average of two weeks per chapter to three or four days a revision. Although the new schedule resulted in many late nights laboring over broken Blender files and settings that changed with each new Blender release, seeing the final result now makes it all worth it ten times over. The skills I developed while writing the book, both as a writer and as a Blender artist, have opened up so many more opportunities for me to grow as a tutor and freelancer.

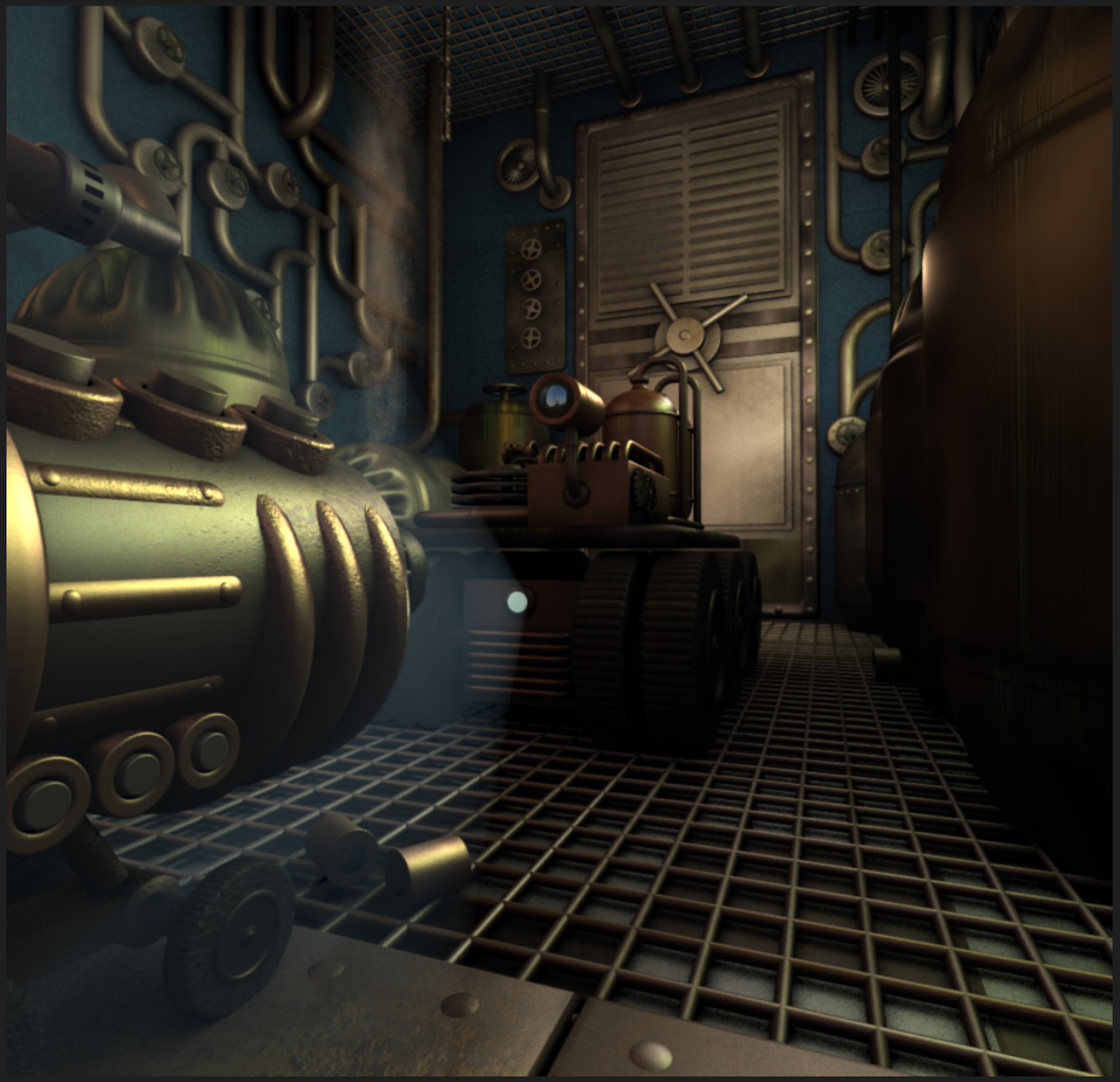
Now that my writing experience with Packt Publishing is over (for now at least), I'm proudly taking what I learned and applying it to my new project, CGShark.com, where I host tutorials written by both myself and other members of the 3D community. Although the tutorial library is small right now, I hope to expand beyond Blender and include tutorials for Autodesk Maya, Adobe Photoshop, Adobe After Effects, and The Gimp. I truly believe that my experience with Packt Publishing directly impacted how I teach students, both online and in person.

If you're interested in learning more about "Blender 2.5 Lighting and Rendering", feel free to visit the official book page on Packt Publishing's website at https://www.packtpub.com/blender-2-5-lighting-and-rendering/book. To download the Blender files used in the book for free, visit the book's page at CGShark.com: http://www.cgshark.com/lighting-and-rendering/.

Father Give Me Legs - by Moises Salvador



Endgame - by Luiz Eduardo Borges



Cosmic Eggs - by reynante

HJ - by Jakub 'Kubeczek77' Batg

Once Upon A Hunted Night - by Bruna Alberto

Sip - by reynante

Ninja Weapon - by Gagan Hajatri

Into The Wind - by Glenn De Backer

Just A Game - by Hans Schwaiger

Want to write for BlenderArt Magazine?

Here is how!

1. We accept the following:

- Tutorials explaining new Blender features, 3dconcepts, techniques or articles based on current theme of the magazine.
- Reports on useful Blender events throughout the world.
- Cartoons related to blender world.

2. Send submissions to sandra@blenderart.org. Send us a notification on what you want to write and we can follow up from there. (Some guidelines you must follow)

- Images are preferred in PNG but good quality JPG can also do. Images should be separate from the text document.
- Make sure that screenshots are clear and readable and the renders should be at least 800px, but not more than 1600px at maximum.
- Sequential naming of images like, image 001.png... etc.
- Text should be in either ODT, DOC, TXT or HTML.
- Archive them using 7zip or RAR or less preferably zip.

3. Please include the following in your email:

- Name: This can be your full name or blenderartist avtar.
- Photograph: As PNG and maximum width of 256Px. (Only if submitting the article for the first time)
- About yourself: Max 25 words .
- Website: (optional)

Note: All the approved submissions can be placed in the final issue or subsequent issue if deemed fit. All submissions will be cropped/modified if necessary. For more details see the blenderart website.

Upcoming Issue 'Theme'

Issue 30

"Under the Microscope"

- Mathematics:
  - Coordinate Systems
  - Integration (animated)
  - Proofing Pythagoras's theorem
- Physics:
  - Animations/Simulations of some mechanic experiments
  - Atom/Subatomic Particles
  - Electronic orbital (visualisation)
- Chemistry:
  - The look of a molecule
- Biology:
  - The animal cell
  - The plant cell
  - Cell division

Disclaimer